IN THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

Please AMEND claims 1, 3, 7, 10, 12 and 22 in accordance with the following:

1. (CURRENTLY AMENDED) A method of calculating and writing a checksum in a memory, comprising:

calculating a first checksum by reading a ~~first~~ set of first values from the memory and summing the read values, and storing the first checksum;

calculating a first mode checksum by subtracting a second ~~set of~~ values written in a predetermined area of the memory from the first checksum;

initializing a second checksum to be zero when the first mode checksum does not meet a predetermined condition;

calculating a second mode checksum by inverting the second checksum, adding the inverted second checksum to the first mode checksum; and

writing the inverted second checksum value in the predetermined area of the memory, when the second mode checksum is equal to the second checksum.

2. (PREVIOUSLY PRESENTED) The method of claim 1, further comprising increasing the second checksum by one and then repeating the calculating of the second mode checksum and the subsequent increasing of the second checksum by one, when the second mode checksum is not the same as the second checksum.

3. (CURRENTLY AMENDED) The method of claim 1, wherein the predetermined condition in initializing the second checksum is a function of modulo calculating the first checksum by a first value to generate a ~~second~~ modulo calculated value.

4. (PREVIOUSLY PRESENTED) The method of claim 3, wherein the initializing the second checksum comprises correcting a bit value located immediately prior to the written area of the inverted second checksum value, when the predetermined condition is met.

5. (PREVIOUSLY PRESENTED) The method of claim 2, wherein the initializing the second checksum comprises correcting a bit value located immediately prior to an entry area of the inverted second checksum value, when the predetermined condition is met.

6. (PREVIOUSLY PRESENTED) The method of claim 1, wherein the initializing the second checksum comprises correcting a bit value located just before the written area of the inverted second checksum value, when the predetermined condition is met.

7. (CURRENTLY AMENDED) The method of claim 1, further comprising inverting and outputting the second value written in the predetermined area when checking the checksum in the memory.

8. (ORIGINAL) The method of claim 1, further comprising selectively displaying at least one of the first checksum, the first mode checksum, the second checksum, and the second mode checksum on a display that is external to the memory.

9. (ORIGINAL) The method of claim 1, wherein the predetermined area of the memory in which the inverted second checksum value is written is a last two bytes of the memory.

10. (CURRENTLY AMENDED) An apparatus for checking a checksum, comprising:

a memory that stores a predetermined program and a checksum;

a microcomputer that executes the predetermined program stored in the memory, reads the checksum stored in the memory, inverts the checksum read from the memory, and outputs the inverted checksum; and

a checksum calculator that calculates and writes the checksum into the memory by:

calculating a first checksum by reading a ~~first~~ set of first values from the memory and summing the read values, and storing the first checksum,

calculating a first mode checksum by subtracting a second ~~set of~~ value~~s~~ written in a predetermined area of the memory from the first checksum,

initializing a second checksum to be zero when the first mode checksum does not meet a predetermined condition,

calculating a second mode checksum by inverting the second checksum and adding the inverted second checksum to the first mode checksum; and

when the second mode checksum is equal to the second checksum, writing the inverted second checksum value in the predetermined area of the memory.

11. (PREVIOUSLY PRESENTED) The apparatus of claim 10, wherein the checksum calculator increases the second checksum by one and repeats the calculating of the second mode checksum and the subsequent increasing of the second checksum by one, when the second mode checksum is not equal to the second checksum.

12. (CURRENTLY AMENDED) The apparatus of claim 10, wherein the predetermined condition is that the result of modulo calculating a first checksum by one of the first values is a ~~second~~ modulo calculated value.

13. (ORIGINAL) The apparatus of claim 12, wherein the checksum calculator corrects a bit value located immediately prior to the predetermined area of the memory on which the inverted second checksum value is written.

14. (ORIGINAL) The apparatus of claim 11, wherein the checksum calculator corrects a bit value located immediately prior to the predetermined area of the memory on which the inverted second checksum value is written.

15. (ORIGINAL) The apparatus of claim 10, wherein the checksum calculator corrects a bit value located immediately prior to the predetermined area of the memory on which the inverted second checksum is written.

16. (ORIGINAL) The apparatus of claim 10, further comprising a display to selectively display at least one of the first checksum, the first mode checksum, the second checksum, the second mode checksum, and the inverted second mode checksum.

17. (ORIGINAL) The apparatus of claim 16, wherein the display is an LED display.

18. (ORIGINAL) The apparatus of claim 16, wherein the display is an on-screen display.

19. (ORIGINAL) The apparatus of claim 10, wherein the memory is a ROM.

20. (ORIGINAL) The apparatus of claim 10, wherein the predetermined area of the memory in which the inverted second checksum value is written is a last two bytes of the memory.

21. (ORIGINAL) The apparatus of claim 10, wherein the checksum is written and checked without dismantling the apparatus.

22. (CURRENTLY AMENDED) A computer readable medium on which a program implementing a method of calculating and writing a checksum in a memory is stored, the program causing a computer to execute the method, wherein the method comprises:

calculating a first checksum by reading a ~~first~~ set of first values from the memory and summing the read values, and storing the first checksum;

calculating a first mode checksum by subtracting a second ~~set of~~ values written in a predetermined area of the memory from the first checksum; and

initializing a second checksum to be zero when the first mode checksum does not meet a predetermined condition;

calculating a second mode checksum by inverting the second checksum and adding the inverted second checksum to the first mode checksum; and

writing the inverted second checksum value in the predetermined area of the memory, when the second mode checksum is equal to the second checksum.

23. (PREVIOUSLY PRESENTED) The computer readable medium of claim 22, wherein the computer readable medium is distributed to a computer system connected through a network and is stored and executed as a computer readable code in a distributed mode.